

3、zigbee 网关单机通信

1. 实验目的

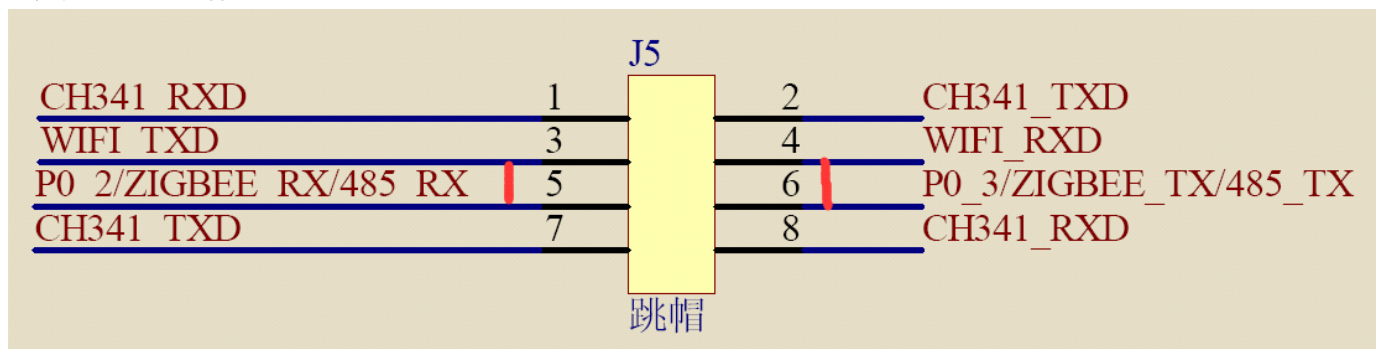
- 1)、通过实验掌握 CC2530 芯片通过串口控制 WIFI 模块的方法
- 2)、掌握 Led 驱动电路及开关 Led 的原理
- 3)、掌握基于 WIFI 的 TCP/IP 无线通信

2. 实验设备

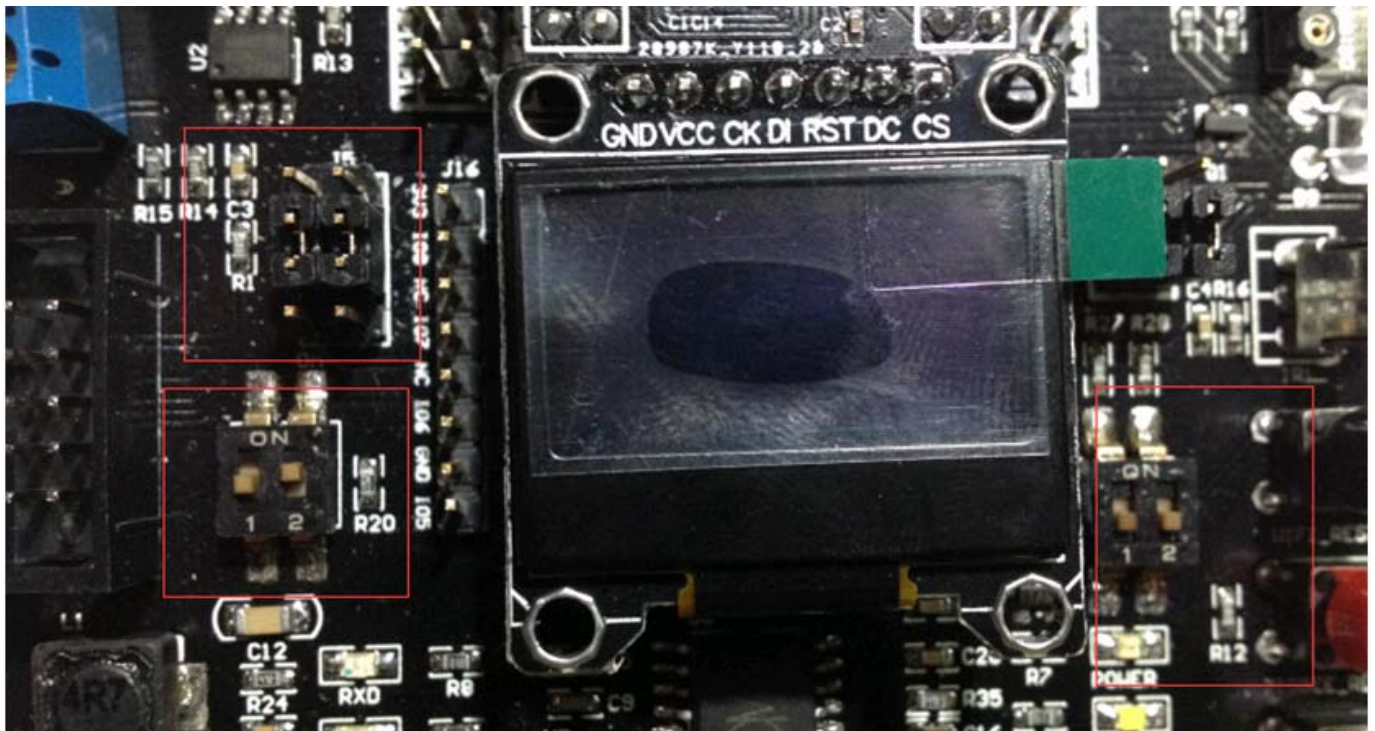
硬件：PC 机一台 ZB2530 网关（底板、核心板、仿真器、USB 线 WIFI 模块）一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

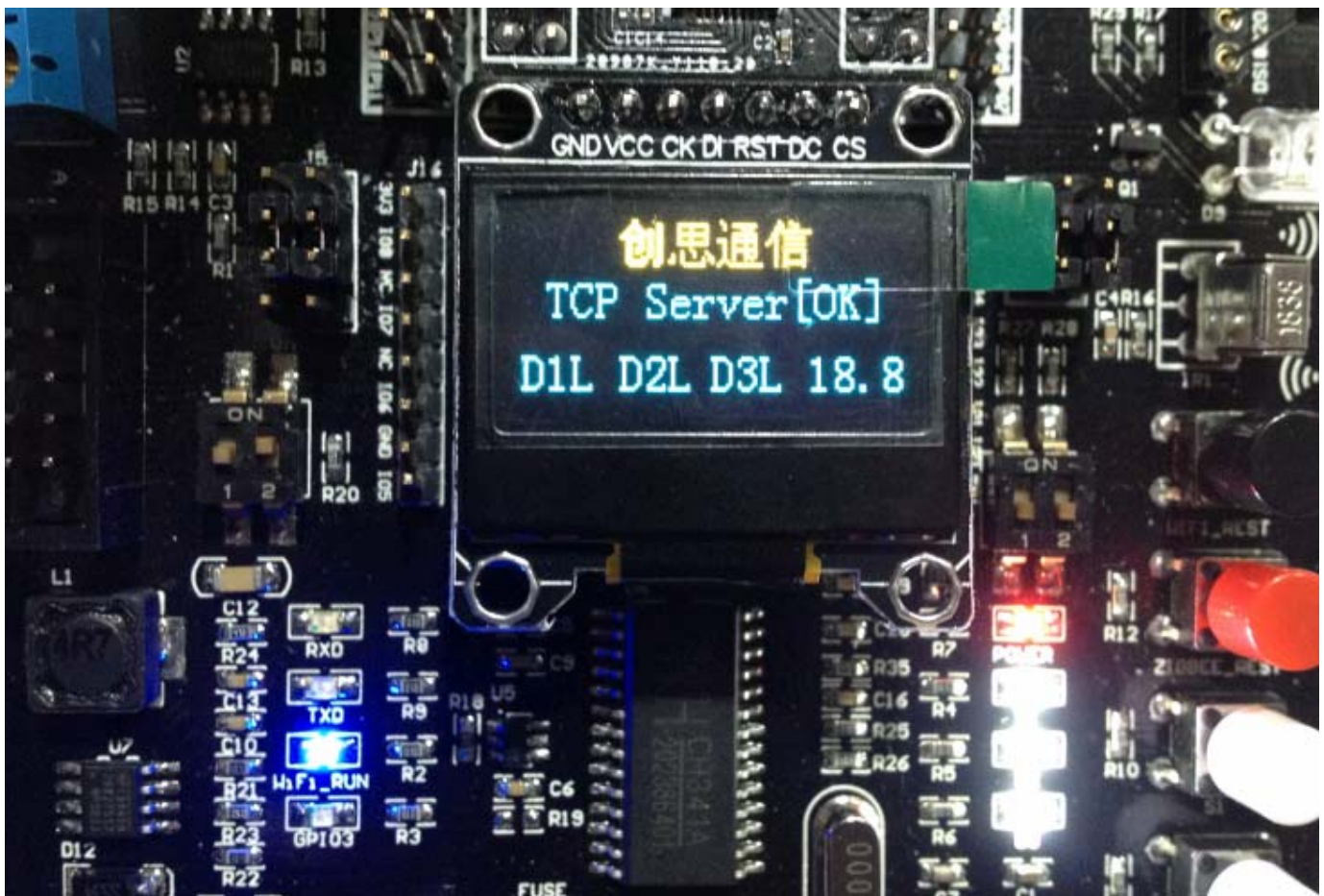
3. 实验相关电路图



如上图 J5 这样接，表明 ZIGBEE 通过串口控制模块。如果 5-7 6-8 则表明 zigbee 模块接入到 PC。如果 1-3 2-4 表明 PC 通过 USB 线[板子自带 USB 转串口芯片 CH341]直接接入 WIFI 模块。



4. 插好液晶屏和温度传感器[也可以不插入液晶屏和温度传感器，DS18B20 数字面朝板子内]



5.硬件初始化

其中 D1 表示 zigbee 芯片和 WIFI 模块进行通信。通信成功则 D2 灯亮起来。

其次 D2 表示 WIFI 模块的服务器初始化，初始化成功则 D3 灯亮起来。

6.代码分析

```
#include <iocc2530.h>
#include <stdio.h>
#include <string.h>
#include "lcd.h"
#include "ds18b20.h"

typedef unsigned char uchar;
typedef unsigned int uint;

#define IGT P0_6    // P0.6 口控制 IGT
uint count;        //用于定时器计数
int g_timeout=0;
/////以下是温度相关
extern void Delay_ms(unsigned int k);//外部函数 ms 的声明
#define ARRAY_SIZE(arr) (sizeof(arr) / sizeof(arr)[0])
#define FLOAT_TEMP
char strTemp[10]; //温度存放

//定义控制灯的端口
#define LED1 P1_0
#define LED2 P1_1
#define LED3 P0_4

void initUART0(void);
void InitialAD(void);
void UartTX_Send_String(uchar *Data,int len);
```

```
#define MAXCHAR 81
uchar Recdata[MAXCHAR];

uchar RXTXflag = 1;
uchar tempRX;
uint datanumber = 0;
uint stringlen;
uint j = 0;
//外部函数
extern void DelayMS(unsigned int msec);

//发送 GPRS 数据
int g_thread = 0;
char g_data[50];
int Send_Wifi_DATA(char *str, int len);
//命令
#define AT_C "AT\r\n"
#define MUX_C "AT+CIPMUX=1\r\n"
#define SERVER_C "AT+CIPSERVER=1,5000\r\n"
#define RST_C "AT+RST\r\n"

//收到数据判断
#define Rev_F "ESP"

#define Rev_D1G "ESPGLED1"
#define Rev_D1K "ESPKLED1"
#define Rev_D2G "ESPGLED2"
#define Rev_D2K "ESPKLED2"
#define Rev_D3G "ESPGLED3"
#define Rev_D3K "ESPKLED3"
```

```
#define Rev_TUL "UNLINUX"  
#define Rev_TLI "LINK"  
#define Rev_GD "GETDATA"  
#define Rev_TUL1 "Unlink"
```

```
void clearBuff(void)  
{  
    for(j=0;j<MAXCHAR;j++)  
    {  
        Recdata[j]=0x00;  
    }  
    j=0;  
}
```

```
/******
```

函数名称：Print_Char

函数功能：发送单个字符

入口参数:ch 出口参数：无

```
*****/
```

```
void Print_Char(char ch)//发送单个字符
```

```
{  
    U0CSR &= ~0x40;          //不能收数  
    U0DBUF = ch;  
    while(UTX0IF == 0);  
    UTX0IF = 0;  
    U0CSR |= 0x40;          //允许接收  
}
```

```
/******
```

函数名称：Print_Str



函数功能：发送字符串

入口参数:*str 出口参数：无

*****/

void Print_Str(char *str)//发送字符串

{

while(*str!='\0')

{

Print_Char(*str);

DelayMS(2);

str++;

}

}

*****/

初始化串口 0 函数

*****/

void initUART0(void)

{

CLKCONCMD &= ~0x40;

//设置系统时钟源为 32MHZ 晶振

while(CLKCONSTA & 0x40);

//等待晶振稳定

CLKCONCMD &= ~0x47;

//设置系统主时钟频率为 32MHZ

PERCFG = 0x00;

//位置 1 P0 口

POSEL = 0x0c;

//P0 用作串口

P2DIR &= ~0XC0;

//P0 优先作为 UART0

U0CSR |= 0x80;

//串口设置为 UART 方式

U0GCR |= 11;

U0BAUD |= 216;

//波特率设为 9600



```
UTX0IF = 1; //UART0 TX 中断标志初始置位 1
```

```
U0CSR |= 0x40; //允许接收
IEN0 |= 0x84; //开总中断，接收中断
}
```

```
/******
```

```
* 名 称: SetWifi()
```

```
* 功 能: 设置 LED 灯相应的 IO 口
```

```
* 入口参数: 无
```

```
* 出口参数: 无
```

```
*****/
```

```
void SetWifi(void)
```

```
{
    PODIR |= 0x40; //P0.6 定义为输出
    IGT = 1; //高电平复位
    DelayMS(500);
    IGT = 0; //低电平工作
}
```

```
void dispIGTState()
```

```
{
    if(IGT>0) //监测 WIFI 模块复位管脚
    {
        //LCD_P8x16Str(8, 5, "IGTH");
    }
    else
    {
        //LCD_P8x16Str(8, 5, "IGTL");
    }
}
```

```
}  
}
```

```
void dispLEDState()  
{  
    if(LED1>0) //监测 WIFI 模块复位管脚  
    {  
        LCD_P8x16Str(0, 5, "D1H");  
        strcat(g_data,"D1H");  
    }  
    else  
    {  
        LCD_P8x16Str(0, 5, "D1L");  
        strcat(g_data,"D1L");  
    }  
  
    if(LED2>0) //监测 WIFI 模块复位管脚  
    {  
        LCD_P8x16Str(32, 5, "D2H");  
        strcat(g_data,"D2H");  
    }  
    else  
    {  
        LCD_P8x16Str(32, 5, "D2L");  
        strcat(g_data,"D2L");  
    }  
  
    if(LED3>0) //监测 WIFI 模块复位管脚  
    {  
        LCD_P8x16Str(62, 5, "D3H");
```



```
    strcat(g_data,"D3H");
}
else
{
    LCD_P8x16Str(62, 5, "D3L");
    strcat(g_data,"D3L");
}
}

void Wifi_Ctrl_ZGBLED(char const * str)
{

    if(strstr(str,Rev_D1G)!=NULL) //监测 WIFI 模块复位管脚
    {
        LED1=1;
    }
    if(strstr(str,Rev_D1K)!=NULL)
    {
        LED1=0;
    }

    if(strstr(str,Rev_D2G)!=NULL) //监测 WIFI 模块复位管脚
    {
        LED2=1;
    }
    if(strstr(str,Rev_D2K)!=NULL)
    {
        LED2=0;
    }

    if(strstr(str,Rev_D3G)!=NULL) //监测 WIFI 模块复位管脚
```

```
{  
    LED3=1;  
}  
if(strstr(str,Rev_D3K)!=NULL)  
{  
    LED3=0;  
}  
}
```

```
void Wifi_Rst(void)  
{
```

```
    clearBuff();  
    Print_Str(RST_C);  
    DelayMS(100);
```

```
    if(strstr((char const *)Recdata,"OK")==NULL)  
    {  
        LCD_P8x16Str(8,2,"Wifi Reset....");  
        DelayMS(5000);  
        clearBuff();  
    }  
    else  
    {  
        LCD_P8x16Str(8,2,"Wifi Reset[OK]");  
        DelayMS(5000);  
        clearBuff();  
    }
```

```
}
```

```
}
```

```
void Connect_Wifi(void)
```

```
{
```

```
clearBuff();
```

```
Print_Str(AT_C);
```

```
DelayMS(50);
```

```
while(strstr((char const *)Recdata,"OK")==NULL)
```

```
{
```

```
LED1=1;
```

```
DelayMS(500);
```

```
LED1=0;
```

```
LCD_P8x16Str(8,2,"connect.....");
```

```
clearBuff();
```

```
Print_Str(AT_C);
```

```
DelayMS(500);
```

```
}
```

```
LCD_P8x16Str(8,2,"connect...[OK]");
```

```
LED1=0;
```

```
}
```



```
void Init_MUX(void)
{
    clearBuff();
    Print_Str(MUX_C);
    DelayMS(50);

    while(strstr((char const *)Recdata,"OK")==NULL)
    {
        LED2=1;
        DelayMS(500);
        LED2=0;
        LCD_P8x16Str(8,2,"Init MUX.....");
        clearBuff();
        Print_Str(MUX_C);
        DelayMS(500);
    }
    LCD_P8x16Str(8,2,"Init MUX..[OK]");
    LED2=0;

}

void Init_Server(void)
{

    clearBuff();
    Print_Str(SERVER_C);
    DelayMS(50);
```



```
while(strstr((char const *)Recdata,"OK")==NULL)
{
    LED3=1;
    DelayMS(500);
    LED3=0;
    LCD_P8x16Str(8,2,"Init Server...");
    clearBuff();
    Print_Str(SERVER_C);
    DelayMS(500);
}
LCD_P8x16Str(8,2,"TCP Server[OK]");
LED3=0;
}
```

```
char Get_DS18B20(void)
{
    float fTemp;
    POSEL &= 0x7f;           //DS18B20 的 io 口初始化

    if(1)                    //插好了温度传感器
    {
        memset(strTemp, 0, ARRAY_SIZE(strTemp));
```

//厂家提供的程序温度值不带小数，Ds18B20 本身是支持 1 位小数位的，修改后使其支持，精度更高

```
#if defined(FLOAT_TEMP)
    fTemp = floatReadDs18B20();    //温度读取函数 带 1 位小数位
    sprintf(strTemp, "%.01f", fTemp); //将浮点数转成字符串
#else
//    ucTemp = ReadDs18B20();        //温度读取函数
//    strTemp[0] = ucTemp/10+48;     //取出十位数
```



```
//    strTemp[1] = ucTemp%10+48;    //取出个位数
    sprintf(strTemp, "%d%d", ucTemp/10, ucTemp%10);
#endif
    LCD_P8x16Str(95, 5, (unsigned char*)strTemp);
    strcat(g_data, "#");
    strcat(g_data, strTemp);
    //Delay_ms(1000);    //延时函数使用定时器方式
}
return 0;
}
```

```
int Send_Wifi_DATA(char *str, int len)
{
    char SEND_C[20];
    memset(SEND_C, 0, 20);
    sprintf(SEND_C, "AT+CIPSEND=%d,%d\r\n", g_thread, len);

    Print_Str(SEND_C);
    DelayMS(50);

    while(strstr((char const *)Recdata, ">") == NULL)
    {
        if(g_timeout > 50)
        {
            g_timeout = 0;
            return -1;
        }
        DelayMS(5);
        g_timeout++;
    }
}
```



```
Print_Str(str);  
return 0;  
}
```

```
/******
```

主函数

```
*****/
```

```
void main(void)  
{  
    Delay_ms(10);          //让设备稳定  
    SetWifi();              //设置 LED 灯相应的 IO 口  
  
    P1DIR |= 0x03;          //P1 控制 LED 10 11 04  
    P0DIR |= 0x10;  
    LED1 = 1;              //关 LED  
    LED2 = 1;              //关 LED  
    LED3 = 1;              //关 LED  
  
    clearBuff();           //清空缓冲  
    initUART0();           //初始化串口  
    Print_Str("init uart [ok]\r\n");  
  
    ////  
    LCD_Init();           //LCD 初始化  
    LCD_CLS();            //清屏  
    LCD_welcome();  
    Print_Str("init OLED [ok]\r\n");  
  
    dispIGTState();        //显示 WIFI 是否开启  
    Wifi_Rst();            //wifi 软件复位
```



```
Connect_Wifi();           //zigbee 和 WIFI 通信
DelayMS(1000);

Init_MUX();
DelayMS(1000);

Init_Server();

while(1)
{
    memset(g_data,0,50);
    strcat(g_data,"ZW#");
    dispIGTState();
    dispLEDState();
    Get_DS18B20();
    if(strstr((char const *)Recdata,Rev_F)!=NULL) //接收到了数据
    {
        Wifi_Ctrl_ZGBLED((char const *)Recdata);
    }

    if(strstr((char const *)Recdata,Rev_TUL)!=NULL)
    {
        LCD_P8x16Str(8,2,"PC/MP UnLink .");
        clearBuff();
    }
    if(strstr((char const *)Recdata,Rev_TUL1)!=NULL)
    {
        LCD_P8x16Str(8,2,"PC/MP UnLink .");
    }
}
```

```
}
if(strstr((char const *)Recdata,Rev_TLI)!=NULL)
{
    LCD_P8x16Str(8,2,"PC/MP Link On.");
    clearBuff();
}

if(strstr((char const *)Recdata,Rev_GD)!=NULL) //搜集采集数据
{
    DelayMS(1000);
    Send_Wifi_DATA(g_data,strlen(g_data)) ;
    DelayMS(1000);
}

clearBuff();
DelayMS(10);
}
}
/*****
串口接收一个字符:一旦有数据从串口传至 CC2530,则进入中断，将接收到的数据赋值给变量
temp.
*****/
#pragma vector = URX0_VECTOR
__interrupt void UART0_ISR(void)
{

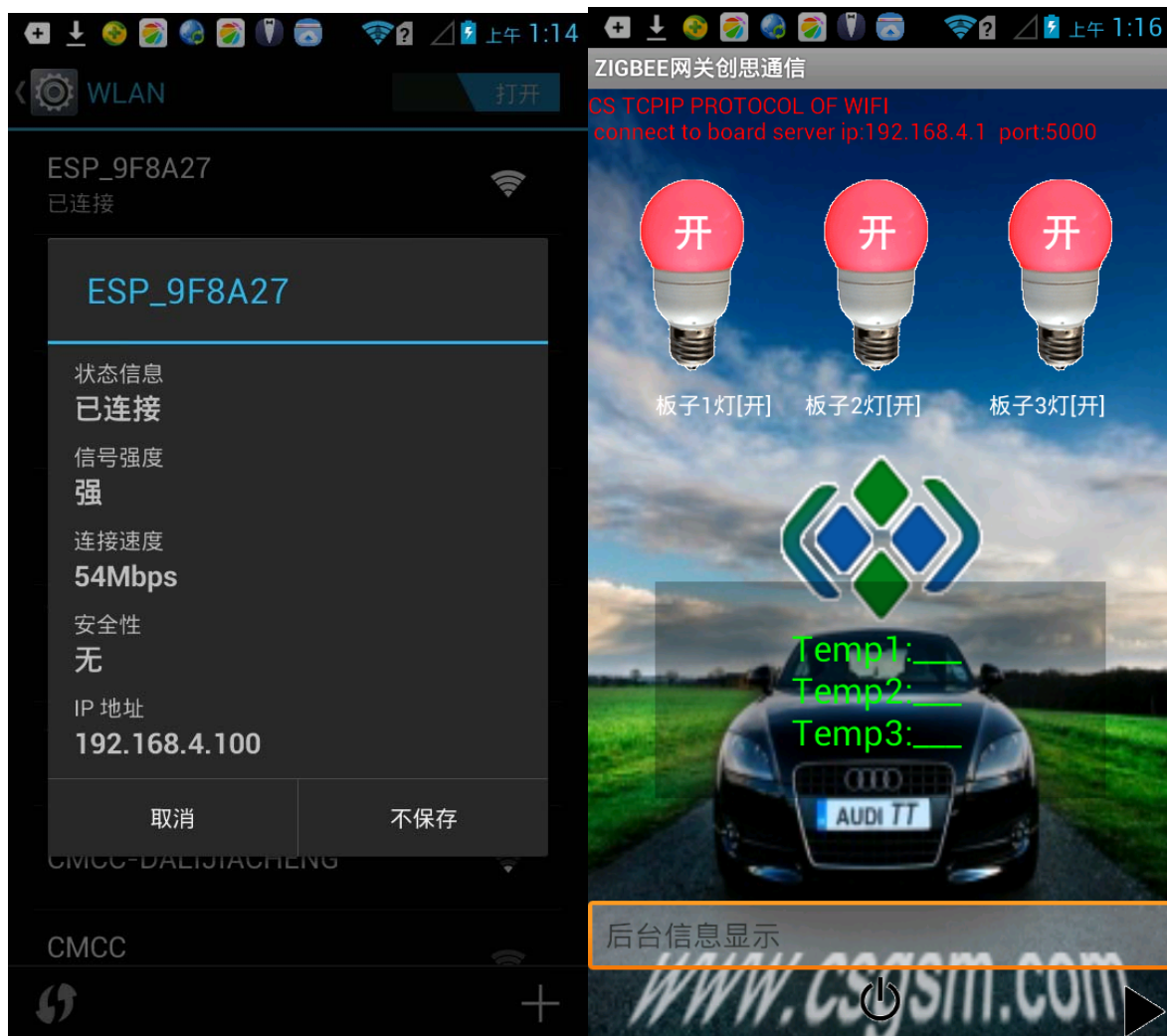
    if(j<81)
    {
        tempRX = U0DBUF;
        Recdata[j]=tempRX;//命令存到命令数组
```

```
URX0IF=0; //软件清除接收中断
j++;
}
}
```

6.实验现象

当把程序下到开发板后，务必给板子重新上电[插拔接入电脑的 USB]。请观察 D1 灯闪烁的情况。如果有 LCD，请同时注意 LCD。

A.手机连接到 WIFI 模块的 AP 热点。然后才打开软件。



B.点击右下脚的三角箭头，启动和板子通信。点击灯泡，关闭和打开开发板上面的 D1 D2 D3 LED 灯

C.温度采集将会在我们手机软件界面上面显示，如果没插入温度传感器[如果要插温度传感器印有DS18B20 的那面朝板子内]。否则会显示 4080.0。

